

FAST TEMPLATE MATCHING BY SUBSAMPLED CIRCULANT MATRIX

Sung-Hsien Hsieh^{*,**}, Chun-Shien Lu^{*}, and Soo-Chang Pei^{**}

^{*}Institute of Information Science, Academia Sinica, Taipei, Taiwan

^{**}Graduate Inst. Comm. Eng., National Taiwan University, Taipei, Taiwan

ABSTRACT

Template matching is widely used for many applications in image and signal processing and usually is time-critical. Traditional methods usually focus on how to reduce the search locations by coarse-to-fine strategy or full search combined with pruning strategy. However, the computation cost of those methods is easily dominated by the size of signal N instead of that of template K . This paper proposes a probabilistic and fast matching scheme, which computation costs requires $O(N)$ additions and $O(K \log K)$ multiplications, based on cross-correlation. The nuclear idea is to first downsample signal, which size becomes $O(K)$, and then subsequent operations only involves downsampled signals. The probability of successful match depends on cross-correlation between signal and the template. We show the sufficient condition for successful match and prove that the probability is high for binary signals with $\frac{K^2}{\log K} \geq O(N)$. The experiments shows this proposed scheme is fast and efficient and supports the theoretical results.

Index Terms— Circulant matrix, Cross-correlation, Sub-sampling, Template matching

1. INTRODUCTION

Template matching is a task of searching a given template in a given signal. It has been widely used in many applications, including communication synchronization [1], quality control [2], compression [3], object detection [4], and etc. Given a template $\mathbf{t} \in \mathbb{R}^K$ and a signal $\mathbf{x} \in \mathbb{R}^N$, template matching is to solve

$$\arg \max_{0 \leq k < N} \text{Sim}(\mathbf{x}^k, \mathbf{t}), \quad (1)$$

where $\mathbf{x}^k = [(\mathbf{x})_k, \dots, (\mathbf{x})_{N-1}, (\mathbf{x})_0, \dots, (\mathbf{x})_{k-1}]$, $(\mathbf{x})_k$ represents the k^{th} entry of \mathbf{x} , and $\text{Sim}(\cdot)$ denotes a kind of similarity metric.

One main challenge of solving Eq. (1) is to incur high computation time but many applications demands real-time and are energy-critical. There are two factors influencing computation overheads. One is what similarity measures, including cross-correlation (CC) [5][6], normalized cross-correlation (NCC) [7][8], and sum of squared differences

(SSD)[9], are used. Similarity measure influences performances as well. Another is the search strategy. For example, exhaustive search (full search) that is conducted by checking \mathbf{x}^k from $k = 0$ to $k = N - 1$, which is unacceptably slow.

Many techniques have been developed to overcome these difficulties as follows.

- Coarse-to-fine strategy [10][11][12]: First, a coarse search is conducted by finding the downsampled template in the downsampled image to yield a good match with less computation overhead. Then, a fine search is conducted in the original space starting from the neighborhood of the best match found in coarse search.
- Full search-equivalent strategy [13][14][15]: It employ rejection schemes derived, where current search is terminated as soon as some criterions are satisfied.
- Fast convolution by Fast Fourier Transform (FFT) [1][6]: For specific similarity measures such CC and SSD, Eq. (1) requires convoluting \mathbf{x} with \mathbf{t} . By convolution theorem, this operation can be quickly done in the frequency domains of \mathbf{x} and \mathbf{t} . In particular, [1] proposing using sparse FFT [16] to replace FFT for further reducing the computation cost.

In this paper, we propose a fast template matching method using downsampled circulant matrix and cross-correlation. For image-based applications, the performance using cross-correlation is usually inferior to that using NCC or SSD. But, for binary signals such as CDMA codes in communications, it indeed works well [1]. Our idea is to search the match for downsampled signal, and no search in the higher resolution space is required. It results in the fact that the computation overhead, which is far less than other approaches, is related to the size of downsampled signal. The crux of our method relies on how to design a circulant matrix to achieve fast template matching and can be considered to be an extension of [1]. The main differences are two-fold: (1) Our idea is based on exploiting the commutative property between circulant matrices instead of adopting sFFT, leading to the advantages that the proposed scheme is more simple and faster than [1]. (2) The computation cost of our method benefits from the size of template, but that of [1] only depends on the size of signal.

2. NOTATIONS

We briefly introduce the notations used in this paper. Let $\mathbf{v} \in \mathbb{R}^N$ be a vector and let $\text{circ}(\mathbf{v})$ be a circulant matrix generated based on the seed vector \mathbf{v} . For example, $\mathbf{V} = \text{circ}(\mathbf{v})$, where the first row of \mathbf{V} is $[(\mathbf{v})_0, (\mathbf{v})_1, \dots, (\mathbf{v})_{N-1}]$, the second row is $[(\mathbf{v})_{N-1}, (\mathbf{v})_0, \dots, (\mathbf{v})_{N-2}]$, and the last row is $[(\mathbf{v})_1, (\mathbf{v})_2, \dots, (\mathbf{v})_0]$. $0_{1 \times (N)}$ represents the $1 \times N$ zero vector. If $\mathbf{u} \in \mathbb{R}^M$ and $\mathbf{v} \in \mathbb{R}^N$, $[\mathbf{u} \ \mathbf{v}] \in \mathbb{R}^{M+N}$ denotes a vector by concatenating \mathbf{u} and \mathbf{v} . We simplify notations to use $(\mathbf{x})_i = (\mathbf{x})_i \text{ modulo } N$.

3. PROPOSED METHOD

Template matching based on cross-correlation is equivalent to solving $m = \arg \max_k (\mathbf{T}\mathbf{x})_k$ with $(\mathbf{T}\mathbf{x})_k = \sum_{i=0}^{K-1} (\mathbf{t})_i (\mathbf{x})_{k+i}$, where $\mathbf{T} = \text{circ}([\mathbf{t} \ 0_{1 \times (N-K)}])$. Specifically, $(\mathbf{T}\mathbf{x})_k$ is considered as the matching result at position k of \mathbf{x} and m is considered as the ground truth with the maximum correlation, $(\mathbf{T}\mathbf{x})_m$. The goal of the proposed method is to find an index \hat{m} , which is expected to be equal to the ground truth m , with fast computation.

The key idea is to downsample an original signal \mathbf{x} via a sampling matrix Φ to become a low-dimensional signal \mathbf{y} , i.e., $\mathbf{y} = \Phi\mathbf{x}$, and then find a matrix $\hat{\mathbf{T}} \in \mathbb{R}^{M \times M}$ such that $\hat{\mathbf{T}}\mathbf{y} = \hat{\mathbf{T}}\Phi\mathbf{x} = \Phi\mathbf{T}\mathbf{x}$. Since, compared with \mathbf{x} , \mathbf{y} has lower dimension, the computation cost of $\hat{\mathbf{T}}\mathbf{y}$ is lower than that of $\mathbf{T}\mathbf{x}$. In other words, $\hat{\mathbf{T}}\mathbf{y}$ is equivalent to fast convolving \mathbf{x} with \mathbf{T} . Moreover, $\Phi\mathbf{T}\mathbf{x}$ is considered to downsample $\mathbf{T}\mathbf{x}$, which is a vector representing matching results based on cross-correlation. Nevertheless, downsampling (i.e., for $\Phi\mathbf{T}\mathbf{x}$) also leads to the side effect that the matching result with the maximum cross-correlation cannot be identified intuitively from $\hat{\mathbf{T}}\mathbf{y}$. To overcome the difficulty, we first require at least two downsampled signals for conducting template matching and then by Chinese Remainder Theorem (CRT) the ground truth can be correctly identified.

The proposed algorithm is depicted in Algorithm 1, which is mainly composed of four operations:

1. Downsampling: Downsample original signal \mathbf{x} into low-dimensional signal \mathbf{y} , where each entry of \mathbf{y} is the sum of $\lceil \frac{N}{M} \rceil$ entries in \mathbf{x} (Step 2-3).
2. Fast convolution in low-dimensional space: Convolve \mathbf{y} with \mathbf{t} and obtain the result \mathbf{r} (Step 4-6). By commutative property between two circulant matrices, \mathbf{y} being convolved by \mathbf{t} is equivalent to \mathbf{x} being convolved by \mathbf{t} . Each entry of the convolution result \mathbf{r} is transformed as the sum of $\lceil \frac{N}{M} \rceil$ entries in $\mathbf{T}\mathbf{x}$.
3. Matching position finding: If an entry in \mathbf{r} is large, it implies that one of $\lceil \frac{N}{M} \rceil$ entries in $\mathbf{T}\mathbf{x}$ is also large

Algorithm 1 Fast Template Matching

Input: $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{t} \in \mathbb{R}^K$; **Output:** \hat{m} ;

01. **function** FTM()
02. Pick two co-prime integers M_1 and M_2 as downsampling factors such that $M_1 M_2 > N$, $M_1 \geq K$, and $M_2 \geq K$;
03. Downsample \mathbf{x} into \mathbf{y}_{M_1} and \mathbf{y}_{M_2} , where
$$(\mathbf{y}_{M_1})_k = \sum_{i=0}^{\lceil \frac{N}{M_1} \rceil - 1} (\mathbf{x})_{k+iM_1} \text{ for } k = 0, \dots, M_1 + K - 1,$$

$$(\mathbf{y}_{M_2})_k = \sum_{i=0}^{\lceil \frac{N}{M_2} \rceil - 1} (\mathbf{x})_{k+iM_2} \text{ for } k = 0, \dots, M_2 + K - 1;$$
04. Assign $\mathbf{t}_{M_1} = [\mathbf{t} \ 0_{1 \times M_1}]$ and $\mathbf{t}_{M_2} = [\mathbf{t} \ 0_{1 \times M_2}]$;
05. $\tilde{\mathbf{y}}_{M_1} = FFT(\mathbf{y}_{M_1})$, and $\tilde{\mathbf{y}}_{M_2} = FFT(\mathbf{y}_{M_2})$;
 $\tilde{\mathbf{t}}_{M_1} = FFT(\mathbf{t}_{M_1})$, and $\tilde{\mathbf{t}}_{M_2} = FFT(\mathbf{t}_{M_2})$;
06. $\mathbf{r}_{M_1} = IFFT(\tilde{\mathbf{t}}_{M_1} * \tilde{\mathbf{y}}_{M_1})$, $\mathbf{r}_{M_2} = IFFT(\tilde{\mathbf{t}}_{M_2} * \tilde{\mathbf{y}}_{M_2})$;
 where “ $*$ ” denotes pixel-wise multiplication.
07. $\tilde{m}_1 = \arg \max_{i \leq M_1} (\mathbf{r}_{M_1})_i$, $\tilde{m}_2 = \arg \max_{i \leq M_2} (\mathbf{r}_{M_2})_i$;
08. $\mathbf{S}_{M_1} = \left\{ k | k = \tilde{m}_1 + i * M_1 \text{ for } i = 0, \dots, \left\lceil \frac{N}{M_1} \right\rceil - 1 \right\}$,
 $\mathbf{S}_{M_2} = \left\{ k | k = \tilde{m}_2 + i * M_2 \text{ for } i = 0, \dots, \left\lceil \frac{N}{M_2} \right\rceil - 1 \right\}$;
09. $\hat{m} = \mathbf{S}_{M_1} \cap \mathbf{S}_{M_2}$;
10. **end function**

with high probability. Thus, by searching the maximum value in \mathbf{r} (Step 7), it provides the information, a set of $\lceil \frac{N}{M} \rceil$ candidate positions, including m .

4. Best matching position choice: Based on the Chinese Remainder Theorem, we can further identify the best template matching result as the unique position \hat{m} (Step 8-9).

The main operations are discussed in detail as follows. We will use M to denote the downsampling factor if there is no confusion.

3.1. Downsampling

Let \mathbf{y} be a downsampled signal defined as:

$$(\mathbf{y})_k = \sum_{i=0}^{\lceil \frac{N}{M} \rceil - 1} (\mathbf{x})_{k+iM} = (\Phi\mathbf{x})_k \quad (2)$$

with

$$\Phi = \mathcal{D}_M(\text{circ}(\mathbf{r})), \quad (3)$$

where $\mathcal{D}_M(\cdot)$ is the function that outputs the first M rows of its argument and \mathbf{r} is generated as follows: $(\mathbf{r})_k = 1$ if $k \bmod M = 0$ for $0 \leq k \leq N - 1$.

There are two useful properties about Φ : 1) $\Phi\mathbf{x}$ only involves additions and 2) Φ is a circulant matrix. It should be noted that M controls the dimension of a downsampled signal. Furthermore, the proposed method requires $M \geq K$; otherwise, there will be no matrix satisfying $\hat{\mathbf{T}}\mathbf{y} = \Phi\mathbf{T}\mathbf{x}$ (details will be discussed later in Theorem 1 and Theorem 2).

3.2. Fat convolution in low-dimensional space

After downsampling, we find a matrix $\hat{T} \in \mathbb{R}^{M \times M}$ such that $\hat{T}\Phi x = \Phi T x$. If this strategy is feasible, it reduces the computation cost since the dimension of \hat{T} is smaller than that of T . However, the problem is not intuitive but difficult in that it is not simply related to the commutative property in matrices. Valsesia and Magli [17] propose Theorem 1, stating under what sufficient conditions the commutative property holds.

Theorem 1. (slightly revised from [17]) Let $T = \text{circ}([t \ 0_{1 \times (N-K)}]) \in \mathbb{R}^{N \times N}$ be a circulant matrix, where t is a $1 \times K$ ($K \leq M$) non-zero vector. Let $\hat{T} \in \mathbb{R}^{M \times M} = \text{circ}([t \ 0_{1 \times (M-K)}])$, let Φ be a $M \times N$ partial circulant matrix, and let $y = \Phi x$ and $\Phi T x$ denote the measurements of x and measurements of filtered signal (Tx), respectively. Then,

$$(\hat{T}y)_i = (\Phi T x)_i \text{ if and only if } i \in [0, M - K].$$

In fact, Theorem 1 also implies that there are K mismatches, namely $(\hat{T}y)_i \neq (\Phi T x)_i$ for $M - K < i \leq M - 1$. These mismatches may lead to the failure of the proposed algorithm. For example, $(\hat{T}y)_k$, in fact, is the sum of $\lceil \frac{N}{M} \rceil$ entries of Tx . If k is the best match position involving $(Tx)_m$, and $(\hat{T}y)_k = (\Phi T x)_k$ (without mismatches), it

means $(\hat{T}y)_k = (Tx)_m + \eta$, where $\eta = \sum_{i=0, i \neq \hat{i}}^{\lceil \frac{N}{M} \rceil - 1} (Tx)_{k+iM}$

and $m = k + \hat{i}M$. We can anticipate that $(\hat{T}y)_k$ may be large enough since it involves $(Tx)_m$. On the contrary, if $(\hat{T}y)_k \neq (\Phi T x)_k$ (with mismatches), $(\hat{T}y)_k$ is unpredictable since $(\hat{T}y)_k = (Tx)_m + \eta$ no longer holds. Under the circumstance, the proposed algorithm loses the information about $(Tx)_m$.

To deal with this problem, we consider two strategies: 1) Instead of using \mathcal{D}_M , we use \mathcal{D}_{M+K} in Eq. (3) to produce Φ . In other words, according to Theorem 1, $(\hat{T}y)_k = (\Phi T x)_k$ holds for $0 \leq k \leq M - 1$. It should be noted that the set of candidate positions collected from $(\hat{T}y)_k$ for $0 \leq k \leq M - 1$ already includes the information of $(Tx)_0, \dots, (Tx)_{N-1}$. Thus, no information is lost. However, it results in more overheads since the size of y is increased to $M + K$. 2) To further reduce such overheads, we derive another sufficient condition without mismatch by designing a new sampling matrix Φ .

Theorem 2. Suppose $\frac{N}{M}$ is an integer. Let $T = \text{circ}([t \ 0_{1 \times (N-M)}]) \in \mathbb{R}^{N \times N}$ be a circulant matrix, where t is a $1 \times M$ non-zero vector, and let $\hat{T} \in \mathbb{R}^{M \times M} = \text{circ}(t)$. We also let $\Phi = \underbrace{[\Phi_M \ \Phi_M \dots \Phi_M]}_{\frac{N}{M}}$ be an $M \times N$ matrix, where Φ_M is an $M \times M$ circulant matrix. Then, we have the measurements $y = \Phi x$ and measurements $\Phi T x$ of filtered signal Tx . Therefore, we have

$$(\hat{T}y)_i = (\Phi T x)_i \text{ if and only if } i \in [0, M - 1].$$

Proof. If both $A \in \mathbb{R}^{N \times N}$ and $B \in \mathbb{R}^{N \times N}$ are circulant matrices, then matrix multiplication is commutative, namely $AB = BA$. In our case, $\hat{T}y = \hat{T}\Phi x = \hat{T}[\Phi_M \ \Phi_M \dots \Phi_M]x = [(\hat{T}\Phi_M) \ (\hat{T}\Phi_M) \dots (\hat{T}\Phi_M)]x$. Since both Φ_M and \hat{T} are circulant matrices, $\hat{T}\Phi_M = \Phi_M \hat{T}$. Thus,

$$\hat{T}\Phi x = [(\Phi_M \hat{T}) \ (\Phi_M \hat{T}) \dots (\Phi_M \hat{T})]x = \Phi \bar{T}x,$$

$$\text{where } \bar{T} = \begin{bmatrix} \hat{T} & 0 & \dots & 0 \\ 0 & \hat{T} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{T} \end{bmatrix}. \quad \square$$

Theorem 2 holds when M divides N . Otherwise, we can pad zeros into the tail of x until M divides N .

3.3. Matching position finding

Let \tilde{y} be the convolved signal with $\tilde{y} = \hat{T}y$. By Theorem 1 or Theorem 2, \tilde{y} can be rewritten as:

$$(\tilde{y})_k = \sum_{i=0}^{\lceil \frac{N}{M} \rceil - 1} (Tx)_{k+iM}. \quad (4)$$

If $\tilde{m} = \arg \max_{k < M} (\tilde{y})_k$, we are interested in the question that whether the ground truth m belonging to the candidate position set \mathbb{C} , which is $\mathbb{C} = \{k | k = \tilde{m} + i * M \text{ for } i = 0, \dots, \lceil \frac{N}{M} \rceil - 1\}$. If yes, the proposed algorithms correctly finds \mathbb{C} to include m . We examine the sufficient condition of successful trials in Theorem 3.

Theorem 3. Let $(Tx)_k = \sum_{i=0}^{K-1} (t)_i (x)_{k+i}$. Let the desired matching result be $m = \arg \max_k (Tx)_k$. If $\frac{1}{2\lceil \frac{N}{M} \rceil + 1} (Tx)_m > \max_{k \neq m} (Tx)_k$, then the proposed algorithm determines \tilde{m} such that $m \in \mathbb{C}$.

Proof. Without loss of generality, we assume $m = 0$, implying $\tilde{m} = 0$ such that $m \in \{k | i * M \text{ for } i = 0, \dots, \lceil \frac{N}{M} \rceil - 1\}$. If $(\tilde{y})_0 > \max_{i \neq 0} (\tilde{y})_i$, the proposed algorithm correctly picks $\tilde{m} = 0$.

To show the sufficient condition of $(\tilde{y})_0 > \max_{i \neq 0} (\tilde{y})_i$, we first derive the lower bound of $(\tilde{y})_0$, which is

$$\begin{aligned} (\tilde{y})_0 &= (Tx)_0 + \sum_{i=1}^{\lceil \frac{N}{M} \rceil - 1} (Tx)_{iM} \\ &\geq (Tx)_0 - \left(\left\lceil \frac{N}{M} \right\rceil - 1 \right) \max_{k \neq 0} (Tx)_k. \end{aligned}$$

Then, the upper bound of $\max_{i \neq 0}(\tilde{\mathbf{y}})_i$ is derived to be $\max_{i \neq 0}(\tilde{\mathbf{y}})_i \leq \lceil \frac{N}{M} \rceil \max_{k \neq 0}(\mathbf{T}\mathbf{x})_k$. Thus, we have

$$(\mathbf{T}\mathbf{x})_0 - (\lceil \frac{N}{M} \rceil - 1) \max_{k \neq 0}(\mathbf{T}\mathbf{x})_k > \lceil \frac{N}{M} \rceil \max_{k \neq 0}(\mathbf{T}\mathbf{x})_k$$

$$\Rightarrow \frac{1}{2\lceil \frac{N}{M} \rceil + 1}(\mathbf{T}\mathbf{x})_0 > \max_{k \neq 0}(\mathbf{T}\mathbf{x})_k.$$

We complete the proof. \square

Although Theorem 3 provides the sufficient condition, the successful probability is still unknown. In fact, the probability is related to the signal type of x . We will give a more detailed analysis taking the practical application as an example later.

3.4. Best matching position

The operation ‘‘Matching position finding’’ yields a set of candidate positions, but the unique position still is unknown. The following discussion is based on the prerequisite that the set of candidate positions includes the correct solution m .

To identify the correct solution, it is equivalent to solving an unknown variable m such that $m \equiv \hat{m} \pmod{M}$. The problem is efficiently solved by Chinese Remainder Theorem as follows.

Theorem 4. (Chinese Remainder Theorem (CRT)) Any integer m is uniquely specified mod N by its remainders modulo α relatively prime integers $M_1, M_2, \dots, M_\alpha$ as long as $\prod_{i=1}^\alpha M_i \geq N$.

More specifically, we consider two co-prime integers in this paper. Let M_1 and M_2 be relatively prime integers such that $M_1 M_2 \geq N$. Hence, the two equations, $m \equiv \hat{m}_1 \pmod{M_1}$ and $m \equiv \hat{m}_2 \pmod{M_2}$, where \hat{m}_1 and \hat{m}_2 are the results obtained from the third operation ‘‘Matching position finding,’’ have the unique solution m that is the best matching position.

3.5. Practical Applications

In this section, we discuss the practical applications using the proposed algorithm. Among them, synchronization is a critical issue in communications. For example, global positioning system (GPS) consumes 30%-75% power for synchronization. Thus, it is crucial to develop cost-effective GPS synchronization. The problem is defined as follows.

- In the sender, a spreading code $\mathbf{x} \in \{1, -1\}^N$, which is also known by the receiver, is sent to the receiver.
- The receiver obtains the delayed code:

$$\mathbf{x}^m = [(\mathbf{x})_m, \dots, (\mathbf{x})_{N-1}, (\mathbf{x})_0, \dots, (\mathbf{x})_{m-1}].$$

- By comparing the spreading code ($\mathbf{t} = \mathbf{x}$ in this paper) and the delayed code, the receiver solves $\hat{m} = \arg \max_{0 \leq k < N} \text{Sim}(\mathbf{x}^k, \mathbf{t})$.

It should be noted that whatever the size of \mathbf{t} is, the computation complexity of fast convolution by FFT is invariant. In other words, even though the size of template is smaller than N , it cannot reduce the computation cost.

Nevertheless, the proposed method is not necessary to set $\mathbf{t} = \mathbf{x}$ for synchronization and can benefit from low-dimensional template. In fact, it suffices to set $\mathbf{t} = [(\mathbf{x})_0, \dots, (\mathbf{x})_{K-1}]$ with the size $K \leq M < N$. We prove in the following theorem the successful probability and computation complexity of our method.

Theorem 5. Let $\alpha = \min(M_1, M_2)$ and let $\beta = \max(M_1, M_2)$. If $\mathbf{x} \in \{1, -1\}^N$, where 1 and -1 have equal probability, and $\mathbf{t} = [(\mathbf{x})_m, \dots, (\mathbf{x})_{m+K-1}]$ for any m , then our proposed algorithm perfectly identifies the matching result \hat{m} to be equal to the correct position m , i.e., $\hat{m} = m$, with the successful probability being larger than $1 - 2 \left(O(\alpha e^{-\frac{K}{\alpha}}) + O(\frac{1}{K} \lceil \frac{N}{\alpha} \rceil) \right)$ and computation costs of $O(N + \beta \log \beta)$ for additions and $O(\beta \log \beta)$ for multiplications.

Proof. The proposed algorithm can be roughly divided into two matching problems based on M_1 and M_2 . Only when both matching problems succeed, the proposed method is considered to be successful. In the following, we first take the matching problem based on M_1 as an example. Definitely, the analysis is also applied to another matching problem involving M_2 .

The probability that the matching problem employing M_1 reports an incorrect result is equal to

$$P_F^{M_1} = \Pr \left[(\hat{\mathbf{T}}\mathbf{y})_m \leq \max_{i \neq m} (\hat{\mathbf{T}}\mathbf{y})_i \right].$$

Since $(\mathbf{T}\mathbf{x})_k = \sum_{i=0}^{K-1} (\mathbf{t})_i (\mathbf{x})_{k+i}$, we have

- If $k = m$, $(\mathbf{T}\mathbf{x})_m = \sum_{i=0}^{K-1} (\mathbf{x})_{m+i}^2 = K$. Furthermore, $E[(\mathbf{T}\mathbf{x})_m] = K$ and $\text{Var}[(\mathbf{T}\mathbf{x})_m] = 0$.
- If $k \neq m$, $E[(\mathbf{T}\mathbf{x})_k] = \sum_{i=0}^{K-1} E[(\mathbf{t})_i (\mathbf{x})_{k+i}] = 0$ due to the independence between $(\mathbf{t})_i$ and $(\mathbf{x})_{k+i}$. Furthermore, both $(\mathbf{t})_i (\mathbf{x})_{k+i}$ and $(\mathbf{t})_j (\mathbf{x})_{k+j}$ are independent for $i \neq j$. Thus, $\text{Var}[(\mathbf{T}\mathbf{x})_k] = \sum_{i=0}^{K-1} \text{Var}[(\mathbf{t})_i (\mathbf{x})_{k+i}] = K$.

$$\text{Recall that } (\hat{\mathbf{T}}\mathbf{y})_k = \sum_{i=0}^{\lceil \frac{N}{M_1} \rceil - 1} (\mathbf{T}\mathbf{x})_{k+iM_1}.$$

Let $S_k = \left\{ l \mid l = k + i * M_1 \text{ for } i = 0, \dots, \lceil \frac{N}{M_1} \rceil - 1 \right\}$. Thus,

- If $m \in S_k$, $E[(\hat{\mathbf{T}}\mathbf{y})_k] = \sum_{i=0}^{\lceil \frac{N}{M_1} \rceil - 1} E[(\mathbf{T}\mathbf{x})_{k+iM_1}] = K$. Furthermore, $E[(\mathbf{T}\mathbf{x})_{k+iM_1} (\mathbf{T}\mathbf{x})_{k+jM_1}] = 0$ for

$i \neq j$. It implies $Cov[(\mathbf{T}\mathbf{x})_{k+iM_1}, (\mathbf{T}\mathbf{x})_{k+jM_1}] = 0$

for $i \neq j$. Thus, $Var[(\hat{\mathbf{T}}\mathbf{y})_k] = \sum_{i=0}^{\lceil \frac{N}{M_1} \rceil - 1} Var[(\mathbf{T}\mathbf{x})_{k+iM_1}]$

$$\sum_{i \neq j}^{\lceil \frac{N}{M_1} \rceil - 1} Cov[(\mathbf{T}\mathbf{x})_{k+iM_1}, (\mathbf{T}\mathbf{x})_{k+jM_1}] = \left(\left\lceil \frac{N}{M_1} \right\rceil - 1\right)K.$$

- If $m \notin S_k$, $E[(\hat{\mathbf{T}}\mathbf{y})_k] = 0$ and $Var[(\hat{\mathbf{T}}\mathbf{y})_k] = \left\lceil \frac{N}{M_1} \right\rceil K$.

Then, we will bound the probability $P_F^{M_1}$ by the following events: $E_1 : \exists k$ and $m \notin S_k$, s.t. $(\hat{\mathbf{T}}\mathbf{y})_k \geq \frac{K}{2}$; $E_2 : \exists k$ and $m \in S_k$, s.t. $(\hat{\mathbf{T}}\mathbf{y})_k \leq \frac{K}{2}$. If none of the events holds, then the algorithm output is correct. In other words, $P_F^{M_1} = \Pr[E_1] + \Pr[E_2]$.

First, we discuss the probability, $\Pr[E_1]$. We start from the probability that for each k such that $m \notin S_k$, $(\hat{\mathbf{T}}\mathbf{y})_k \geq \frac{K}{2}$. It should be noted that, in this case, $(\hat{\mathbf{T}}\mathbf{y})_k$ is considered as a sum of independent random variables taking values in $\{1, -1\}$ with the probability $\frac{1}{2}$. Thus, for each k , $\Pr[(\hat{\mathbf{T}}\mathbf{y})_k \geq \frac{K}{2}]$ is derived by using Chernoff bound as follows:

$$\Pr\left[(\hat{\mathbf{T}}\mathbf{y})_k \geq \frac{K}{2}\right] \leq e^{-\frac{(\frac{K}{2})^2}{2\lceil \frac{N}{M_1} \rceil}} = e^{-\frac{K}{8\lceil \frac{N}{M_1} \rceil}}.$$

Since the cardinality of $\{k|m \notin S_k\}$ is at most $M_1 - 1$, then

$$\Pr[E_1] \leq (M_1 - 1)e^{-\frac{K}{8\lceil \frac{N}{M_1} \rceil}} < M_1 e^{-\frac{K}{8\lceil \frac{N}{M_1} \rceil}}.$$

Second, $\Pr[E_2]$ is bounded by using Chebyshev's inequality:

$$\begin{aligned} \Pr[E_2] &= \Pr\left[(\hat{\mathbf{T}}\mathbf{y})_k \leq \frac{K}{2}\right] = \Pr\left[(\hat{\mathbf{T}}\mathbf{y})_k \leq K - \frac{K}{2}\right] \\ &= \Pr\left[(\hat{\mathbf{T}}\mathbf{y})_k - K \leq -\frac{K}{2}\right] = \Pr\left[K - (\hat{\mathbf{T}}\mathbf{y})_k > \frac{K}{2}\right] \\ &\leq \frac{Var[(\hat{\mathbf{T}}\mathbf{y})_k]}{(\frac{K}{2})^2} = \frac{4(\lceil \frac{N}{M_1} \rceil - 1)}{K}. \end{aligned}$$

Consequently, $P_F^{M_1} \leq \Pr[E_1] + \Pr[E_2] \leq M_1 e^{-\frac{K}{8\lceil \frac{N}{M_1} \rceil}} + \frac{4(\lceil \frac{N}{M_1} \rceil - 1)}{K}$.

Similarity, the above derivations apply to the second matching problem based on M_2 . It results in $P_F^{M_2} \leq M_2 e^{-\frac{K}{8\lceil \frac{N}{M_2} \rceil}} + \frac{4(\lceil \frac{N}{M_2} \rceil - 1)}{K}$.

In sum, the proposed algorithm succeeds with the probability being larger than $(1 - P_F^{M_1})(1 - P_F^{M_2})$. If $M_1 > M_2$, then $P_F^{M_1} < P_F^{M_2}$. For simplification, let $\alpha = \min(M_1, M_2)$ and we have $(1 - P_F^{M_1})(1 - P_F^{M_2}) \geq (1 - P_F^\alpha)^2 \geq 1 -$

$2\left(O(\alpha e^{-\frac{K}{8\lceil \frac{N}{\alpha} \rceil}}) + O(\frac{1}{K} \lceil \frac{N}{\alpha} \rceil)\right)$. We complete the proof regarding the successful probability.

To prove the computation complexity, we check Algorithm 1 in a step-by-step manner. Step 2 depends on how to pick co-prime integers. In our case, co-prime integers are assigned by setting $M_1 = K$ and $M_2 = K + 1$, and, thus, the cost is negligible. Step 3 obviously costs $O(N)$ additions. Step 4 assigns vectors with the sizes being smaller than M_1 or M_2 and the cost is also negligible. Steps 5 and 6 perform FFT and cost $O(M_1 \log M_1) + O(M_2 \log M_2)$. Step 7 searches the maximum value within the vector with size M_1 or M_2 , and the cost is negligible. Furthermore, in Steps 8 and 9, the cardinality of S_{M_1} and S_{M_2} are $\lceil \frac{N}{M_1} \rceil$ and $\lceil \frac{N}{M_2} \rceil$ respectively. Both of them are smaller than $O(N)$.

Consequently, the computation cost is bounded by Step 3, and Steps 5 and 6 with the total cost being $O(N + M_1 \log M_1 + M_2 \log M_2)$ for additions and $O(M_1 \log M_1 + M_2 \log M_2)$ for multiplications. For simplification, if we let $\beta = \max(M_1, M_2)$, then the cost is bounded by $O(N + \beta \log \beta)$ for additions and $O(\beta \log \beta)$ for multiplications. We complete the proof. \square

4. SIMULATION RESULTS

The simulations were conducted in an Matlab R2012b environment with an Intel CPU Q6600 and 16 GB RAM under Microsoft Win7 (64 bits). We compare the proposed algorithm implemented in C with fast convolution by FFT [6] and sparse FFT [1]. Another goal is also to verify our theoretical analyses.

The testing procedure is:

1. Generate $\mathbf{x} \in \{1, -1\}^N$ and template is extracted from \mathbf{x} with $\mathbf{t} = [(\mathbf{x})_m, \dots, (\mathbf{x})_{m+K-1}]$.
2. Input \mathbf{x} and \mathbf{t} into Algorithm 1 and output \hat{m} , where $M_1 = K$ and $M_2 = K + 1$.
3. If $m = \hat{m}$, the trial is successful. Otherwise, it is failed.

We repeats this procedure 1000 times and calculates the successful probability.

In Fig. 1, we verify the computation cost derived in Theorem 5. Fast convolution by FFT is considered as the baseline. It should be noted that all the results shown in Fig. 1 have successful probabilities fixed at 100%. In Fig. 1(a), the results for our method are shown by fixing N at 2^{26} with K being increased from 2^{16} to 2^{25} . However, it should be noted that the complexities of [6] and [1] only depend on N . Thus, their computation costs are invariant to K . On the other hand, the proposed algorithm benefits from the smaller size of template. One can observe that the proposed method outperforms [6] and [1] when $\frac{N}{K} \geq 2^2$ and $\frac{N}{K} \geq 2^4$, respectively, implying that our algorithm is efficient when the template size is

(far) smaller than the corresponding signal size. Furthermore, when $K \leq 2^{21}$, the computation cost is, in fact, dominated by $O(N)$ instead of $O(M \log M)$ and increased slowly. On the contrary, when $O(M \log M)$ dominates the computation cost, which will be about twice larger when K is doubled.

In Fig. 1(b), the results for our method are shown by fixing K with N being increased from 2^{16} to 2^{26} . In addition, the results for so-called “Proposed Method (best)” are shown, where N still is increased from 2^{16} to 2^{26} but, for each N , K is assigned as small as possible such that successful probabilities achieves 100% based on Theorem 5. Overall, these results indicate that (1) Our method outperforms [6] and [1] especially when N is large enough, implying that, for a large-scale problem, it will be more efficient. (2) When $N \leq 2^{21}$, small sizes of templates (for dot-curve) suffice to quickly achieve 100% successful template matching. In a similar way to Fig. 1(a), when $N \geq 2^{21}$, $O(N)$ overwhelms $O(M \log M)$. In this case, the slopes of red solid and dot curves (our methods) approximate to those of solid black (FFTW) and solid blue (sFFT) curves, where both complexities are $O(N)$ and $O(N \log N)$, respectively.

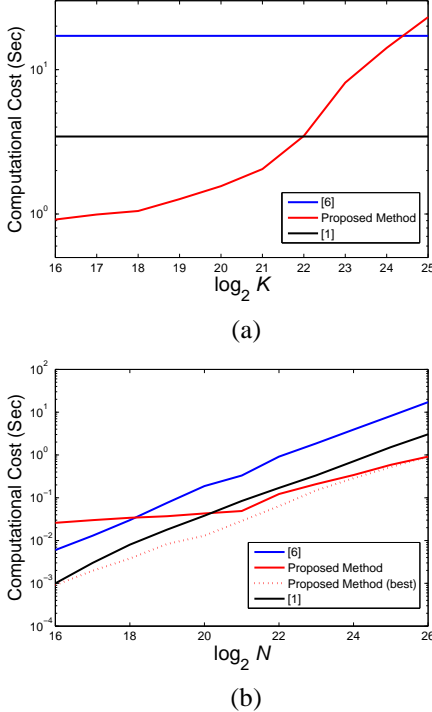


Fig. 1. (a) The computation cost versus different K 's under $N = 2^{26}$. (b) The computation cost versus different N 's under $K = 2^{16}$. All comparisons were conducted with successful probabilities being fixed at 100%

Table 1 further verifies the theoretical probability derived in Theorem 5. By Theorem 5, the proposed algorithm succeeds with the probability being larger than 1 –

$$2 \left(O(\alpha e^{-\frac{K}{s \lceil \frac{N}{\alpha} \rceil}}) + O(\frac{1}{K} \lceil \frac{N}{\alpha} \rceil) \right). \text{ In our case, } \alpha = \min(M_1, M_2),$$

where M_1 and M_2 , in fact, were set to K and $K + 1$, respectively. Thus, when the condition $K^2 \geq O(N)$ holds,

$\frac{1}{K} \lceil \frac{N}{\alpha} \rceil = \frac{1}{K} \lceil \frac{N}{K} \rceil \rightarrow 0$. Similarly, we have $\alpha e^{-\frac{K}{s \lceil \frac{N}{\alpha} \rceil}} \rightarrow 0$ with $\frac{K^2}{\log K} \geq O(N)$. In this experiment, $\frac{N}{K}$ was fixed as 2^{10} . The results show that along with the increase of N , the condition $\frac{K^2}{\log K} = \frac{N^2}{2^{20} \log N - c} \geq O(N)$ (c is a positive constant), which is equivalent to $\frac{N}{\log N} \geq O(2^{20})$, holds. Under the condition, Algorithm 1 succeeds with high probability, as depicted in Table 1.

Table 1. Successful probabilities under $K = \frac{N}{2^{10}}$.

| N | 2^{23} | 2^{24} | 2^{25} | 2^{26} | 2^{27} |
|------------------|----------|----------|----------|----------|----------|
| Successful Prob. | 0.05 | 0.21 | 0.85 | 1 | 1 |

Finally, we test whether the proposed method is robust to noisy inference. In this case, let $x_e = x + e$, where $e \in \mathbb{R}^N$ is additive Gaussian random noise. Both x_e and t were fed into Algorithm 1. The parameters, $N = 2^{26}$ and $M = 2^{16}$, were chosen because the corresponding successful probability lives on edge between 100% and $< 100\%$, which is expected to be interfered by noise obviously. One can observe from Table 2 that our method works well when SNRs are larger than 1 dB.

Table 2. Successful probabilities under $N = 2^{26}$, $K = 2^{16}$, and different SNRs.

| SNR (dB) | 20 | 10 | 6 | 1 | -2 | -6 |
|------------------|----|----|------|------|------|------|
| Successful Prob. | 1 | 1 | 0.95 | 0.72 | 0.48 | 0.06 |

5. CONCLUSIONS AND FUTURE WORK

We present a fast and cost-effective template matching scheme in this paper. We exploit the commutative property of partial circulant matrix to design the sensing matrix for template matching. Our theoretical analyses and simulation results show that the proposed method outperforms FFT- and sparse FFT-based methods. The future work will be examining fast template matching with similarity measures other than cross-correlation.

6. ACKNOWLEDGMENT

This work was supported by Ministry of Science and Technology, Taiwan (ROC), under grants MOST 104-2221-E-001-019-MY3 and 104-2221-E-001-030-MY3.

7. REFERENCES

- [1] H. Hassanieh, P. Indyk, D. Katabi, and Eric Price, "Faster gps via the sparse fourier transform," in *ACM MOBICOM*, 2012.
- [2] M. S. Aksoy, O. Torkul, and I. H. Cedimoglu, "An industrial visual inspection system that uses inductive learning," *Journal of Intelligent Manufacturing*, vol. 15, pp. 569–574, 2004.
- [3] T. Luczak and W. Szpankowski, "A suboptimal lossy data compression based on approximate pattern matching," *IEEE Transactions on Information Theory*, vol. 43, pp. 1439–1451, 1997.
- [4] R. M. Dufour, E. L. Miller, and N. P. Galatsanos, "Template matching based object recognition with unknown geometric parameters," *IEEE Transactions on Image Processing*, vol. 11, pp. 1385–1396, 2002.
- [5] P. E. Anuta, "Spatial registration of multispectral and multitemporal digital imagery using fast fourier transform," *IEEE Transactions on Geoscience Electronics*, vol. 8, pp. 353–368, 1970.
- [6] J. P. Lewis, "Fast template matching," in *Proc. Vision Interface*, 1995, pp. 120–123.
- [7] and S.-H. Lai S.-D. Wei, "Fast template matching based on normalized cross correlation with adaptive multilevel winner update," *IEEE Transactions on Image Processing*, vol. 17, pp. 2227–2235, 2008.
- [8] W. H. Pan, S. D. Wei, and S. H. Lai, "Efficient ncc-based image matching in walsh-hadamard domain," in *Proc. European Conf. Computer Vision: Part III*, 2008, pp. 468–480.
- [9] S. Santini and R. Jain, "Similarity measures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 871–883, 1999.
- [10] G. J. VanderBrug and A. Rosenfeld, "Two-stage template matching," *IEEE Transactions on Computing*, vol. 26, pp. 384–393, 1977.
- [11] M. Gharavi-Alkhansari, "A fast globally optimal algorithm for template matching using low-resolution pruning," *IEEE Transactions on Image Processing*, vol. 10, pp. 526–533, 2001.
- [12] A. Mahmood and S. Khan, "Correlation-coefficient-based fast template matching through partial elimination," *IEEE Transactions on Image Processing*, vol. 21, pp. 2099–2018, 2012.
- [13] S. Mattoccia, F. Tombari, and L. D. Stefano, "Fast full-search equivalent template matching by enhanced bounded correlation," *IEEE Transactions on Image Processing*, vol. 17, pp. 528–538, 2008.
- [14] H. Schweitzer, R. Deng, and R. F. Anderson, "A dual-bound algorithm for very fast and exact template matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 459–470, 2011.
- [15] W. Quyang, F. Tombari, S. Mattoccia, L. D. Stefano, and W.-K. Cham, "Performance evaluation of full search equivalent pattern matching algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 127–143, 2012.
- [16] H. Hassanieh, P. Indyk, D. Katabi, and Eric Price, "Nearly optimal sparse fourier transform," *STOC*, 2012.
- [17] D. Valsesia and E. Magli, "Compressive signal processing with circulant sensing matrices," in *IEEE international conference on Acoustic, Speech and Signal Processing*, 2014, pp. 1015–1019.